*TREXIN INSIGHT PAPER*

May 24, 2023

# USER ACCEPTANCE TESTING IN AGILE DEVELOPMENT

*Successfully adapting your User Acceptance Testing into an Agile development environment requires careful consideration and analysis of your company's unique situation.*

A year ago, we published a Trexin Insight Paper (TIP) comparing User Acceptance Testing (UAT) and System Integration Testing (SIT), where we explained why they are separate processes and should not be completed at the same time, this TIP can be found here. Considering how popular that TIP was, we wanted to revisit the subject and expand on how UAT works in an Agile development environment; something we only briefly mentioned previously. Although in theory UAT should be completed before the end of every sprint, reality often requires adjustments to better match business needs, with organizations typically pursuing three different paths to control costs and manage complexity.

*If you are familiar with UAT in a Waterfall development environment, and Agile feel free to skip the next three sections.*

## USER ACCEPTANCE TESTING REFRESHER

User Acceptance Testing is the process of ensuring that a software solution meets the business requirements (needs) it was designed for. The ultimate question being asked by UAT is: "Is this useful for the end user". At the end of UAT, the end-users should be satisfied with the software; established through multiple pre-arranged tests where a representative sample of users use various aspects of the software within a testing environment. The formal documented approval process usually consists of a decision as to whether the software is "good to go" or not based on the results of the tests. If the software is deemed "good to go", then it's handed off to release testing and management where training will take place, and the software will begin being fully used by end-users.

## USER ACCEPTANCE TESTING IN WATERFALL

Waterfall, traditional, and software development is when software is developed in a purely linear fashion with each phase of development starting when the previous one is sufficiently complete. Waterfall projects are traditionally planned out with a Gantt chart. This means that the entire development, and feature set, is extensively planned from the start providing a firm idea of when the software will be complete and how much it will cost. However, Waterfall has increasingly fallen out of favor as it is highly rigid, not allowing the feature set to be changed following post-planning feedback or market changes without delaying every step afterwards. In a Waterfall development environment, all UAT is done at the same time once the product team has ensured that the software meets its technical specifications. User Acceptance Testing is the last step before the product begins implementation.

## AGILE DEVELOPMENT

The Agile methodology seeks to improve from Waterfall development by viewing software projects through the lens of continuous improvement. Development is chopped into sprints, typically ranging from 1-4 weeks long; during a sprint the product and development team: plans which features they can add to the product, develops them, tests them, and adds them to the product, and then reviews how the sprint went as a whole before starting the next sprint. The goal is that each sprint should build a minimally viable product (MVP) which incorporates the sprint's features, this is an iterative development cycle. The planned features (user stories) for future sprints are continually re-prioritized, added,

and removed from a document called the *Product Backlog* by a person known as the *Product Owner*. If you are curious to learn more about Agile development, more details can be found here.

## USER ACCEPTANCE TESTING IN AGILE

User Acceptance Testing in an Agile development process should be part of what are called User Stories. User Stories are features that are planned to be added to the product and should be written from the perspective of an End User (i.e. as a Finance Analyst I want to be able to easily view the company's current and historical revenue from the reports page and download it into excel), they also contain the Acceptance Criteria which explain when the feature is complete and the User Story is finished. The Acceptance Criteria are once again written from the perspective of the End User and should go through everything which must be true for the feature to be considered successful (ie I should be able to select any date range from today to 10 years ago, I should not be able to download any data besides the revenue, etc.). User Acceptance Testing in an Agile environment should consist of ensuring that all of the Acceptance Criteria for a given User Story have been met. Note that just like in a Waterfall development, UAT should still be performed last after the technical specifications have been met and tested (ie Unit Testing (UT) and System Integration Testing (SIT)).

This means that in theory UAT should be completed towards the end of each sprint of an Agile development system, supporting one of the core ideas of Agile, which is to fully complete a given set of User Stories each sprint and produce an MVP. Enabling a process of continuous improvement and requirements re-prioritization. However, it is common for the practical implementation of Agile to differ substantially from theory, when it comes to UAT this is typically due to two reasons:

1. **Project size and complexity**: In a highly complex project where many features are interrelated it may not be feasible to fully test the impact of a newly added user story on the final product during each sprint due to the time involved with doing this. The feature must be developed and then tested against its technical specifications before it can be moved to UAT and if the product is sufficiently complex it can be unrealistic to expect UAT to fully finish in only a couple of days.
2. **Cost:** Fully testing software is an expensive process, and not every company can spare the resources to comprehensively test their software every couple of weeks.

There are three common solutions to these two issues:

1. Move to a model where larger scope UAT is performed at the end of epics (large workflows comprising multiple sprints) and only more basic UAT is performed at the end of individual sprints. This may mean a whole sprint(s) is dedicated to UAT at the end of each epic.
2. A project where basic UAT is performed at the end of sprints, as in solution one, but as the product approaches release the epics may become more UAT focused; instead of having later sprints at the end of each epic being more UAT focused.
3. User Acceptance Testing is performed entirely at the end of the product's development cycle similar to what is done in a Waterfall development environment. This solution has significant downsides as it means that one of the primary benefits of Agile is not taken advantage of. Without UAT it is difficult to ensure that the product is continually course corrected to meet user needs and that the product backlog is optimal.

Not every one of these three common solutions works in every situation, and they each have their own advantages and disadvantages; implementing Agile can be downright risky to an organization's performance and bottom line, especially if it is done incorrectly. Generally, each organization and project will have to tailor their solutions to meet their own

needs. However, they do not have to do so alone, the assistance of an expert can be instrumental in ensuring that your organization maximizes its return.

If you are curious about how our Advisors at Trexin can help you, please contact us here.

This TIP was written by Duncan Goeden. Duncan welcomes comments and discussion on this topic and can be reached at duncan.goeden@trexin.com.